International Journal for Modern Trends in Science and Technology
Volume 12, Special Issue 01, pages 269-273.
ISSN: 2455-3778 online
Available online at: http://ijmtst.com/ICRCSET2026.html
DOI: https://doi.org/10.5281/zenodo.19561844

MAEG: A Multi-Agent Framework for Intelligent Constraint-Aware Exam Paper Generation

Tejas Sutar, Vedant Talekar, Akash Yadav, Jay Patil, Shilpali Bansu

Department of Artificial Intelligence and Data Science, A.C. Patil College of Engineering, Kharghar, Navi Mumbai, Maharashtra, India

To Cite this Article
Tejas Sutar, Vedant Talekar, Akash Yadav, Jay Patil & Shilpali Bansu (2026). MAEG: A Multi-Agent Framework for Intelligent Constraint-Aware Exam Paper Generation. International Journal for Modern Trends in Science and Technology, 12(SI01), 269-273. https://doi.org/10.5281/zenodo.19561844

Article Info
Received: 02 March 2026; Revised: 01 April 2026; Accepted: 04 April 2026.

KEYWORDS

Question Paper Generation, Multi-Agent Systems, Bloom's Taxonomy, Knowledge Graph, LangGraph, Retrieval-Augmented Generation, Educational AI

ABSTRACT

Question paper generation is a time-intensive task in higher education, often leading to inconsistencies in difficulty, cognitive balance, and syllabus coverage. Existing automated approaches rely on rigid pipelines or single-agent large language models (LLMs), resulting in poor constraint handling and structural instability. This paper presents MAEG, a modular multi-agent framework for constraint-aware university exam paper generation. MAEG employs a three-phase pipeline: (i) input analysis with knowledge graph construction and Bloom's taxonomy-based labeling of previous year questions (PYQs); (ii) blueprint generation using a planner–evaluator agent loop for iterative refinement; and (iii) question selection via a retrieval-first strategy with validation and targeted repair. Evaluated against human-authored papers and a direct LLM baseline, MAEG achieves an average expert score of 43.0/45, with significant improvements in cognitive-level distribution and question usability.

I. INTRODUCTION

Assessment design is central to evaluating student learning outcomes in higher education. University-level question paper generation requires balancing syllabus coverage, cognitive complexity as defined by Bloom's Taxonomy [1], difficulty distribution, and institutional constraints. In practice, this process is performed manually by faculty, often leading to inconsistencies in difficulty, topic weightage, and cognitive-level alignment [2].

To address these challenges, we propose MAEG, a multi-agent framework for constraint-aware exam paper generation. MAEG decomposes the generation process into three phases: (i) input analysis with knowledge graph construction and Bloom's-aligned PYQ structuring; (ii) blueprint generation via a planner–evaluator agent loop; and (iii) question selection using a retrieval-first strategy followed by verification and targeted repair.

Our key contributions are:

- A blueprint-first generation paradigm with immutable structural constraints.
- A multi-agent planner–evaluator loop for iterative refinement with deterministic repair.
- A retrieval-first strategy for efficient and grounded question selection.
- Question-level validation and repair ensuring cognitive and structural consistency.

The system is implemented using LangGraph [3] and FastAPI [4], and uses GPT-4o-mini [5] for cost-efficient inference at $0.01–$0.02 per paper.

II. LITERATURE SURVEY

Manual question paper creation is a time-intensive process requiring balance across exam patterns, Bloom's taxonomy, syllabus coverage, and institutional constraints, often resulting in quality inconsistencies [2]. Bloom's Taxonomy [1] and its revised formulation [6] provide the standard framework for cognitive complexity, yet achieving consistent cognitive distribution across a complete exam remains challenging.

A. Automated Question Generation

Early AQG methods relied on NLP parsing and cloze-based transformations [7], generating isolated questions without structural control. Rule-based and ontology-driven systems improved control but lacked flexibility [8]. Recent transformer-based models and LLMs [5] enable high-quality generation, but direct prompting produces poor constraint handling and structural incoherence.

B. Multi-Agent Systems

Multi-agent systems have been applied in adaptive testing [9], primarily for delivery rather than structured generation. Frameworks such as LangGraph [3] enable iterative stateful agent workflows, though their application to exam generation remains limited.

C. Retrieval-Augmented Generation

RAG [10] improves factual grounding by combining retrieval with generation. Its use in structured exam generation—particularly for leveraging PYQs—is underexplored.

Summary: **Existing approaches lack integrated support for structural planning, constraint enforcement, cognitive control, and efficient reuse of prior question data.**

Summary: Existing approaches lack integrated support for structural planning, constraint enforcement, cognitive control, and efficient reuse of prior question data.

III. RESEARCH GAPS

- **Lack of structural planning:** Questions are generated without a global blueprint, causing inconsistencies in format and coverage.
- **Coarse cognitive control:** Bloom's taxonomy is applied at a high level, limiting fine-grained question-level cognitive distribution.
- **Limited constraint handling:** Current systems do not effectively enforce both institutional requirements and user-defined preferences.
- **Inefficient regeneration:** Full-paper regeneration for error correction results in high cost and instability.
- **Underutilization of PYQs:** Most systems generate questions from scratch, ignoring valuable prior question data.

IV. METHODOLOGY

MAEG decomposes question paper generation into three phases, orchestrated via a LangGraph-based multi-agent workflow.

A. Phase 1: Input Analysis

This phase converts raw inputs into structured representations for downstream processing.

Syllabus Structuring: The syllabus is parsed into modules, topics, subtopics, course outcomes, and weightage.

Knowledge Graph: A graph is constructed with nodes representing syllabus entities and edges capturing part-of, prerequisite, and related-to relationships, supporting coverage analysis and topic substitution during repair.

PYQ Processing: PYQs are stored as structured objects with attributes: text, topic, marks, bloom_level, and metadata. Topic-wise Bloom's distribution and repetition frequency are computed to guide blueprint feasibility.

Constraint Extraction: Teacher inputs are parsed into hard constraints (strict requirements) and soft preferences (guidance signals), with exam structure and Bloom's distribution formalized for blueprint generation.

B. Phase 2: Blueprint Generation and Evaluation

Blueprint Planner: Generates a structured blueprint specifying section, topic, marks, bloom_level, question_type, and is_pyq for each slot.

Critic-Evaluator: Assesses the blueprint against marks consistency, pattern compliance, coverage, Bloom's deviation, PYQ feasibility, and constraint satisfaction.

Iterative Refinement: Planner and evaluator operate in a feedback loop (max 3 iterations). If unmet, the best blueprint is deterministically repaired to ensure convergence. The finalized blueprint is immutable; downstream stages operate within its constraints.

C. Phase 3: Question Selection and Verification

Question Selection: Each slot is filled using a retrieval-first strategy: (i) exact PYQ match, (ii) topic-based retrieval with rephrasing, (iii) semantic retrieval with adaptation, (iv) generation if no match exists.

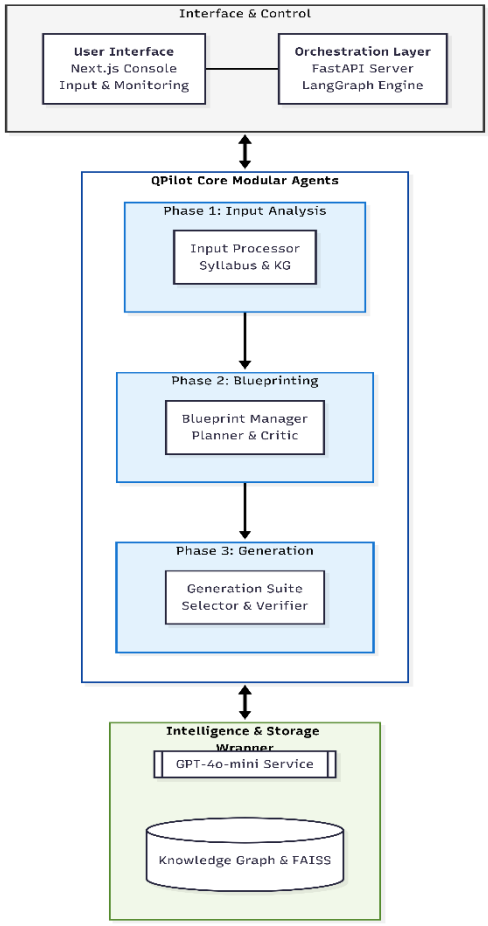
Verification and Repair: Questions are validated for Bloom's alignment, duplication, phrasing quality, and marks–difficulty consistency. Only flagged questions are regenerated via targeted repair, preserving valid content. The final paper is exported using ReportLab.

V. SYSTEM ARCHITECTURE AND ALGORITHMS

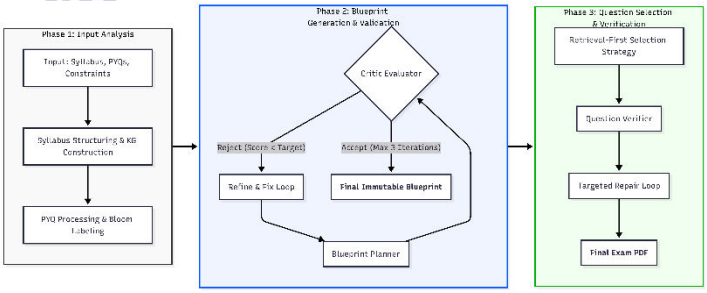
B. System Architecture

MAEG follows a modular pipeline with a Next.js frontend and FastAPI backend hosting the LangGraph multi-agent workflow. Agent interactions are coordinated through a shared state object updated at each node. The stack includes GPT-4o-mini for LLM reasoning, sentence-transformers with FAISS for semantic retrieval, and ReportLab for PDF generation.

Fig. 1. MAEG system architecture



C. Pipeline Flow



D. Data Schemas

- **PYQ Object:** {text, topic, marks, bloom_level, metadata}
- **Blueprint Slot:** {section, topic, marks, bloom_level, question_type, is_pyq}
- **Question Object:** {text, source, marks, bloom_level}

VI. Results and Discussion

A. Experimental Setup

MAEG was evaluated on five university-level subjects using syllabus documents, PYQs, and predefined exam patterns. Three approaches were compared: Human (manually authored), ChatGPT (single-prompt LLM baseline), and MAEG (proposed system). Expert evaluation used five criteria scored out of 9 each: Content Alignment, Cognitive Level Distribution, Difficulty Balance, Time Adequacy, and Question Usability (total: 45).

B. Subject-wise Evaluation

TABLE I: Subject-wise Expert Evaluation Scores (Out of 45)

Subject	Human
Deep Learning	39
Project Management	40
Data Mining	41
Software Engineering	41
Artificial Intelligence	40
Average	**40.2**

C. Average Metric Scores

Metric	Human
Content Alignment	9.0
Cognitive Distribution	7.6
Difficulty Balance	8.0
Time Adequacy	7.4
Question Usability	8.0
Total Score	**40.2**

TABLE II: Average Expert Evaluation Metrics

D. Key Findings

- MAEG achieved the highest scores across all evaluated subjects.
- Significant improvements in cognitive distribution (9.0 vs 7.6 Human) and question usability (8.8 vs 8.0 Human).
- Performance gains were strongest in technical subjects: Deep Learning, Data Mining, and Artificial Intelligence.
- ChatGPT produced fluent outputs but showed weaker structural balance and lower usability.

E. Discussion

MAEG consistently outperformed both baselines in overall exam quality. The blueprint-driven multi-agent approach produces more balanced and exam-ready papers, with the most notable improvements in cognitive-level distribution and usability, validating the effectiveness of the planner–evaluator loop. The retrieval-first strategy reduced repetition and improved question clarity. Cost efficiency at $0.01–$0.02 per paper indicates practical feasibility for real-world deployment.

VII. Conclusion

This paper presented MAEG, a multi-agent framework for constraint-aware university exam paper generation across three phases: input analysis, blueprint generation and evaluation, and question selection with verification. MAEG demonstrates that structured blueprint-driven agent workflows improve structural consistency, cognitive alignment, and efficiency over holistic LLM-based generation. Future work includes large-scale evaluation across diverse subjects, integration of lightweight Bloom's classifiers, and automated answer key generation.

Acknowledgment

The authors thank Prof. S. P. Bansu for guidance throughout this project, the Department of Artificial Intelligence and Data Science at A.C. Patil College of Engineering for infrastructure support, and Mumbai University faculty who contributed evaluation feedback.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] B. S. Bloom et al., Taxonomy of Educational Objectives, Handbook I: Cognitive Domain. New York: David McKay, 1956.

[2] A. Kurdi et al., "A systematic review of automatic question generation for educational purposes," Int. J. AI in Education, vol. 30, no. 1, pp. 121–204, 2020.

[3] LangChain, "LangGraph," 2024. [Online]. Available: https://langchain-ai.github.io/langgraph/

[4] S. Ramírez, "FastAPI," 2024. [Online]. Available: https://fastapi.tiangolo.com/

[5] OpenAI, "GPT-4o technical report," 2024. [Online]. Available: https://openai.com/index/gpt-4o-system-card/

[6] L. W. Anderson and D. R. Krathwohl, A Taxonomy for Learning, Teaching, and Assessing. New York: Longman, 2001.

[7] M. Heilman and N. A. Smith, "Good question! Statistical ranking for question generation," in Proc. NAACL HLT, 2010, pp. 609–617.

[8] C. Papasalouros et al., "Automatic generation of MCQs from domain ontologies," in Proc. e-Learning Conf., 2008.

[9] P. Brusilovsky and C. Peylo, "Adaptive and intelligent web-based educational systems," Int. J. AI in Education, vol. 13, pp. 159–172, 2003.

[10] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Advances in NeurIPS, vol. 33, 2020, pp. 9459–9474

[11] I. N. Alrawashdeh et al., "Antibacterial activity of silver nanoparticles against enteric pathogens," 2019.